Raspberry Pi Recipes Part #3

Let’s get Serial

By **Tony Dixon**
(United Kingdom)

Last time we looked at the Raspberry Pi’s Expansion Header and its General Purpose Input/Ouput (GPIO) signals. In this Elektor **.**Post we’ll take a look at the Raspberry Pi’s Serial UART which can also be found on the Expansion Header.

The RS-232 serial interface can be regarded as the ‘mother of interfaces’. From the early days of computers and the Internet this communication protocol has been around and — luckily — is also available on the Raspberry Pi board. A quick recap on RS-232 pin functions is given in the **inset**.

Serial Interfaces

The Serial UART (Universal Asynchronous Receiver/Transmitter) is one of the three serial interfaces you’ll find on the Raspberry Pi’s Expansion Header. The other two interfaces are an I^{2}C interface and a SPI interface.

Tables 1a-b detail the Expansion Header signals, and the UART can be found on pins 8 (TxD) and 10 (RxD). Unfortunately, the expansion header only has one other UART signal, RTS on pin 11, so you wouldn’t be able to do much in the way of hardware handshaking.

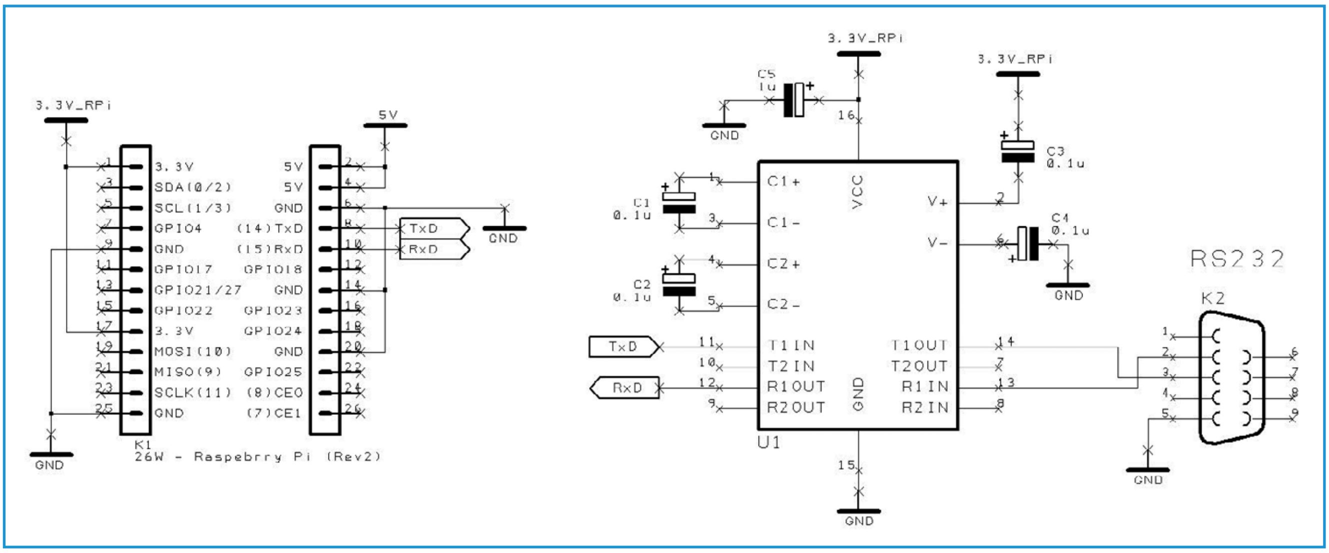
Level up the voltage

First you need an RS-232 interface to convert the 3.3 V UART signal to ±12 V signals used by the RS-232 standard. **Figure 1** shows a simple RS-232 level changer schematic. Here a MAX3232 or equivalent 3.3-V RS 232 transceiver is used to provide us our RS-232 signal levels. **Figure 2** shows the author’s hardware setup where a small commercial add-on board is used to provide the RS232 interface.

Disabling Serial Shell Console

By default the Raspbian distribution allows access to the shell console via the Raspberry

Figure 1.
Schematic for Raspberry Pi RS232 Adapter.



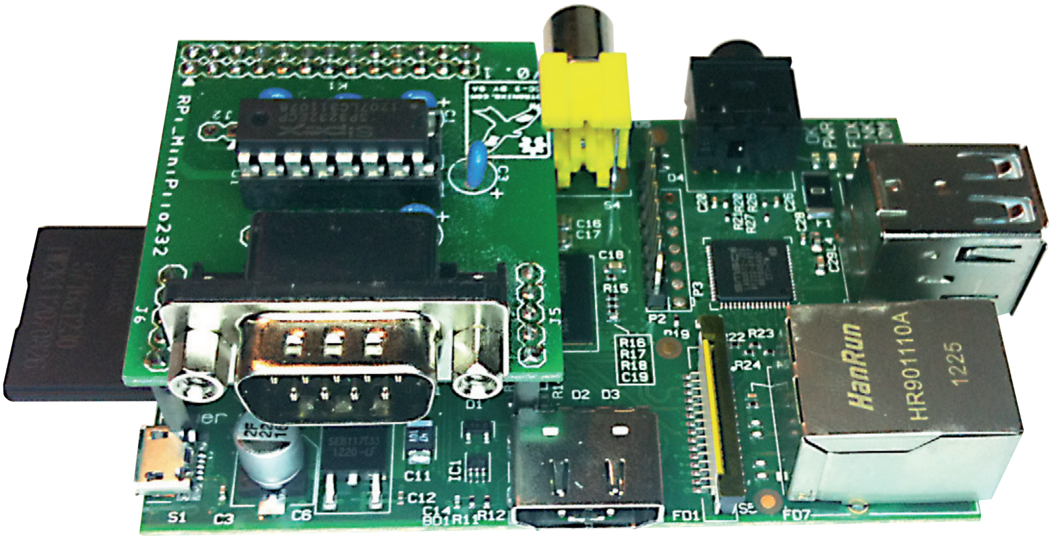


Figure 2.
Pi and RS232 Add-On Board.

Pi's serial UART. This can be extremely useful if you don't have a monitor and keyboard but still need console access your Pi. But this can be a problem for user programs wanting access to the serial port. So you are going to first have to disable this feature. This is easy to do and only requires a few small changes to the *cmdline.txt* and *inittab* files.

Before you do anything, make a backup of the two files you'll be changing.

```
sudo cp /boot/cmdline.txt /boot/
cmdline.bak
sudo cp /etc/inittab /etc/inittab.bak
```

To modify cmdline.txt, in a LXterminal shell type the command:

```
sudo leafpad /boot/cmdline.txt
```

You'll need to find and remove "console=ttyAMA0,115200" and "kgdboc=ttyAMA0,115200" configuration parameters.

The *cmdline.txt* will probably contain the default configuration line:

```
dwc_otg.lpm_enable=0
console=ttyAMA0,115200
kgdboc=ttyAMA0,115200
console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline
rootwait
```

Find and remove "console=ttyAMA0,115200" and "kgdboc=ttyAMA0,115200" configuration parameters so it looks like this:

```
dwc_otg.lpm_enable=0 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

When finished, save the file and quit the editor.

The final step is to edit the "/etc/inittab" file and disable use of the "ttyAMA0" serial port. Type this command:

```
sudo leafpad /etc/inittab
```

At the bottom of the file, look for a configuration line that includes the "ttyAMA0" port address.

Type a hash (or pound) sign ("#") in front of the line "ttyAMA0".

Save the "/etc/inittab" file and exit the editor. You need to reboot our Pi.

Once rebooted, to use the serial port in your own programs you'll use the device address "ttyAMA0" to access the serial port.

Installing Python's Serial Library

As you discovered last time Python is already is installed as standard in the Raspian distribution but to access the Pi's UART we'll need to install a suitable serial hardware library. PySerial is a python library for interfacing with serial interfaces but it does not come as standard with Raspian so we'll

RS-232 Interface pin functions

Pin	Signal	Description	DTE	DCE
1	DCD	Data Carrier Detected	IN	OUT
2	RD (or RxD)	Receive Data	IN	OUT
3	TD (or TxD)	Transmit Data	OUT	IN
4	DTR	Data Terminal Ready	OUT	IN
5	GND	Signal Ground	GND	GND
6	DSR	Data Set Ready	IN	OUT
7	RTS	Ready to Send	OUT	IN
8	CTS	Clear to Send	IN	OUT
9	RI	Ring Indicator	IN	OUT

Table 1a.

Pin Name	Pin Function	Alternative	RPi.GPIO
P1-02	5.0V	-	-
P1-04	5.0V	-	-
P1-06	GND	-	-
P1-08	GPIO14	UART0_TXD	RPi.GPIO8
P1-10	GPIO15	UART0_RXD	RPi.GPIO10
P1-12	GPIO18	PWM0	RPi.GPIO12
P1-14	GND	-	-
P1-16	GPIO23		RPi.GPIO16
P1-18	GPIO24		RPi.GPIO18
P1-20	GND	-	-
P1-22	GPIO25		RPi.GPIO22
P1-24	GPIO8	SPI0_CE0_N	RPi.GPIO24
P1-26	GPIO7	SPI0_CE1_N	RPi.GPIO26

Table 1b.

Pin Name	Board Revision 1		Board Revision 2	
	Pin Function	Alternative	Pin Function	Alternative
P1-01	3.3V	-	3.3V	-
P1-03	GPIO0	I2C0_SDA	GPIO2	I2C1_SDA
P1-05	GPIO1	I2C0_SCL	GPIO3	I2C1_SCL
P1-07	GPIO4	GPCLK0	GPIO4	GPCLK0
P1-09	GND	-	GND	-
P1-11	GPIO17	RTS0	GPIO17	RTS0
P1-13	GPIO21		GPIO27	
P1-15	GPIO22		GPIO22	
P1-17	3.3V	-	3.3V	-
P1-19	GPIO10	SPI0_MOSI	GPIO10	SPI0_MOSI
P1-21	GPIO9	SPI0_MISO	GPIO9	SPI0_MISO
P1-23	GPIO11	SPI0_SCLK	GPIO11	SPI0_SCLK
P1-25	GND	-	GND	-

Note: I2C0_SDA and I2C0_SCL (GPIO0 & GPIO1) and I2C1_SDA and I2C1_SCL (GPIO2 & GPIO3) have 1.8kΩ pull-up resistors to 3V3.

have to download it ourselves and install it. Start an LXterminal session and type the following commands:

```
sudo apt-get install python-serial
```

Like in **Figure 3**.

Example program: serial.py

With pySerial installed you’re now going to write a small test program to send characters to a PC terminal program. Double click the IDLE icon on your Pi’s desktop to start the Python Shell and IDE, see **Figure 4**. Now select File option from the menu and create a new program. This will start the IDE editor, see **Figure 5**. In the IDLE editor, type the program as shown in **Listing 1**.

Once you’ve typed the program, make sure you save it, then switch to an LX Terminal and type the following command to make your program an executable:

```
chmod +x serial.py
```

Once done, you can run your program by typing the following command:

```
sudo ./serial.py
```

If you’ve got a terminal program running on your PC and your PC’s RS232 port connected to your Pi we should see “Hello Elektor” displayed on the terminal screen.

(130151)

Further Reading

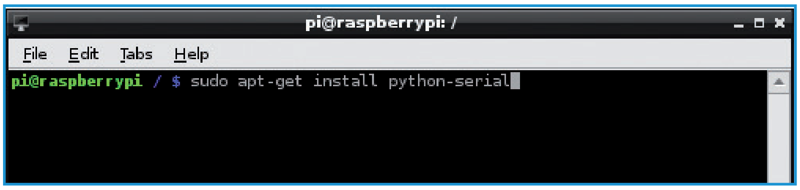
Raspberry Pi:
www.raspberrypi.org

pyserial Library:
https://pypi.python.org/pypi/pyserial

MiniPiio RS232 add-on board:
www.dtronixs.com

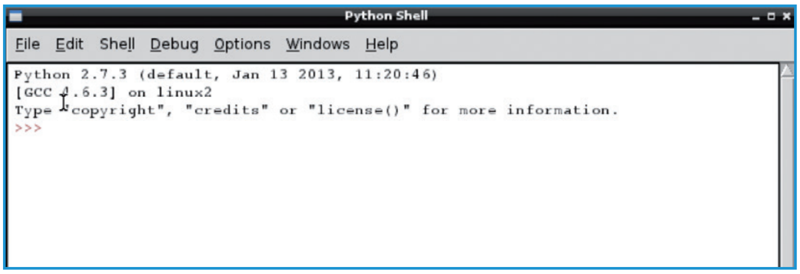
Listing 1.

```
#!/usr/bin/python

import serial

ser = serial.Serial ('/dev/ttyAMA0', 115200, timeout=1)
ser.write ("Hello Elektor")
ser.close ()
```



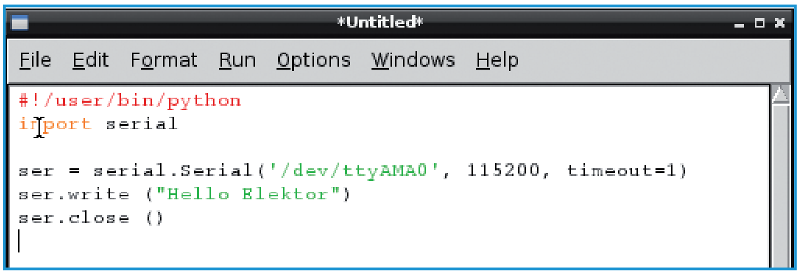
```
pi@raspberrypi / $ sudo apt-get install python-serial
```

Figure 3. LXTerminal in use.



```
Python 2.7.3 (default, Jan 13 2013, 11:20:46)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
```

Figure 4. IDLE Python Shell.



```
#!/user/bin/python
import serial

ser = serial.Serial('/dev/ttyAMA0', 115200, timeout=1)
ser.write ("Hello Elektor")
ser.close ()
```

Figure 5. IDE Editor in action.